Available online at www.sciencedirect.com

ScienceDirect

IERI Procedia 10 (2014) 45 – 50

Procedia IERI

www.elsevier.com/locate/procedia

2014 International Conference on Future Information Engineering

Improvement of the Byzantine Agreement Problem under Mobile P2P Network

Hui-Ching Hsieh[a], Mao-Lun Chiang[b]

[a]*Department of Information Communication, Hsing Wu University, New Taipei City, Taiwan, R.O.C.*
[b]*Department of Information and Communication Engineering, Chaoyang University of Technology, Taichung County, Taiwan, R.O.C.*

Abstract

For improving the accuracy under P2P networks, it must be assured that all non-faulty peers can reach agreement. As such, all the non-faulty peers need to work collaboratively despite disturbances caused by faulty peers. This agreement issue is usually called as the Byzantine Agreement (BA) problem. In previous works, $\lfloor (n\text{-}1)/3 \rfloor + 1$ bouts for exchanging message are necessary to allow all non-faulty peers reaching an agreement. Furthermore, the message complexity of these algorithms are $O(n^n)$. Hence, the relevant algorithms are not suitable for mobile P2P networks in which there may have a great quantity of mobile peers. In this study, a more efficient algorithm has been proposed to decrease the required bouts for exchanging message. Our proposed algorithm only need to run three bouts for exchanging message to allow all non-faulty peers to reach an agreement despite some peers roaming among the different network. It also can decrease the message complexity to $O(n^2)$. It is more suitable and efficient than previous efforts aimed at the mobile P2P network.

Keywords: Byzantine Agreement problem; fault-tolerance; distributed system; mobile P2P network

1. Introduction

Mobile Peer-to-Peer (P2P) networks can be defined as a distributed system comprised of a great quantity of peers. Basically, the accomplishments of mobile P2P network depend upon the ability to spread content efficiently and correctly by utilizing the transmission capacity and cooperation of all peers. Thus, a mobile P2P network has the ability to serve large numbers of peers based on the good service quality [1] under the

doi:10.1016/j.ieri.2014.09.089

indispensable condition that all peers can reach agreement and cooperate well in the network. Unfortunately, in the real situation, there may have some faulty peers to disturb the cooperation between peers. These faulty peers may do some activities to degrade the network performance. Hence, it's important to propose an algorithm to assure the correctness of the network system even if there have some faulty peers.

In the past, there have many fault tolerance schemes been studied, in which the agreement algorithms has attracted much attention. Basically, there have many algorithms [2][3][4][5][7][9][10][11] been proposed for processors to reach agreement or for security applications in a distributed system despite failed processors. This kind of unanimity problem is defined as the Byzantine Agreement problem (BA problem) [5], and was originally proposed by Lamport et al in 1982. Basically, the proposed algorithm must reach the following agreement and validity requirements:

- Agreement: It means that all non-faulty processors need to agree on a unanimity value;
- Validity: It means that the final common value v must equal to the start's initial value, if the starter processor is non-faulty.

Unfortunately, the previous algorithms [6][8][10] require $\lfloor (n\text{-}1)/3 \rfloor + 1$ bouts for exchanging message. Furthermore, the complexity with message will be $O(n^n)$. These algorithms cannot be applied for mobile P2P networks since it may include millions of peers, resulting in a great quantity of overhead while exchanging message.

Besides, peers can roam between different mobile P2P networks without disrupting the execution of the applications arbitrarily. The previous BA algorithms can reach agreement under the pre-defined network topologies [6][8][10]. However, network technology continues to grow quickly and applications in mobile P2P networks recently have reached high complexity. In other words, the previous algorithms cannot make all non-faulty peers to reach an agreement when there have peers roam around the network. Thus, the traditional algorithms [6][8][10] are unsuitable for mobile P2P networks. The agreement problem must be revised under the mobile P2P network.

In order to make the BA algorithm more suitable for mobile P2P network, we revisit and propose a new algorithm: Byzantine Agreement algorithm for Mobile P2P network (BAMP2P), to ensure all peers get an agreement result within three bouts for exchanging message while tolerating the largest quantity of faulty peers.

The rest of this article is presented as follows: The details of BAMP2P are given in Section 2. The correctness is shown in Section 3, and the conclusions are shown in Section 4.

2. The proposed algorithm BAMP2P

In general, each non-faulty peer will execute the same algorithm BAMP2P simultaneously to reach agreement, and BAMP2P includes two phases of works: the exchanging message phase and the making decision phase.

Here, all non-faulty peers must execute three bouts for exchanging message to collect sufficient messages to reach agreement. Noticeably, some peers may roam the network during the execution of BAMP2P. We suppose that all peers can only roam in the network while executing the exchanging message phase. It's due to the reason that there has no message exchange activity during the period of making decision phase. If peers can roam about at that period of time, peers will not have enough messages to reach an agreement correctly. In other words, if the algorithm allows peers roaming about the network during the making decision phase, there will be insufficient messages for peers to determine reliable peers and to reach agreement correctly. Thus, this assumption is needed to assure that all peers have sufficient information to determine reliable peers and to avoid interrupting this process. The BAMP2P procedure is stated below and described by Fig. 1.

Exchanging message Phase:

First, each peer needs to execute three bouts for exchanging message and the collected messages will be stored in the ms-tree of each peer. However, peers have the ability to roam among the different networks during the exchanging message phase. For building the ms-tree correctly, the following roaming conditions must be considered:

- **New peers roam in the network:**

If there has new peer roam in the network, peers originally in the network need to distrubte the values of (r-1)th level to these new peers. These new immigrated peers must apply the majority value on the values and store it to the ms-tree's (r-1) level. Subsequently, the exchanging message phase can be executed continuously.

- **Peers emigrate away from the network:**

If peers move out the network, the peers who still exist in the network must eliminate the messages sent by the absent peers.

Making Decision Phase:

In the making decision phase, peers need to determine which peers are reliable by evaluating the received messages. Then, all non-faulty peers can utilize the replacement process to revise the values which are received from un-reliable peers. Finally, the VOTE function will be applied to the root of the third level in all non-faulty peers' ms-trees, and agreement is reached.

Algorithm BAMP2P

***Exchanging message phase*:**

If *bout* = 1

{

- The starter peer broadcasts it's initial value v_s to others;
- All peers store v_s to the ms-tree's root;

}

For *bout* = 2 to 3

{

- If there has peers roam in the network, then all peers need to execute the function: *immigration-procession* function;
- If peers migrate away the network, then all peers need to execute the function: *processing- emigrate* function;
- Send the values at level (r-1)th within the ms-tree to others in the network.
- Stores the received values to the ms-tree's rth level.

}

***Making decision phase*:**

- Determining the reliable peers:

For the sub-trees of vertex $v(ax)$ within the 2'nd level of the ms-tree

{

If ($v(ax)=maj_{\text{sib-3}}(\text{ancestor}_{ax})$ && $\#\, maj_{\text{sib-3}}(\text{ancestor}_{ax}) \geq (n-\lfloor(n-1)/3\rfloor-1)$)

{

Join peer x into RLP_x;

For all vertexes within the 3'rd level of the ms-tree

{

If ($v(axy) = v(ax)$)

Join peer y to RLP_x ;

}

}

```
        }
        For each peer z (denoted as p_z)
        {
                Total #p_z from all RLP;
                If (#p_z ≥ (n-⌊(n-1)/3⌋))
                {
                        Peer z is reliable;
                }
        }
```

- The replacement process:

```
        For each vertex within the 3'rd level of the ms-tree
        {
                If (peer y is not a reliable peer && v(axy) ≠ maj_sib-3-RP(ancestor_ax))) then
                        v(axy) = maj_sib-3-RP(ancestor_ax);
        }
```

- After the replacement process, the vertices which are repeating in the ms-tree must be deleted.
- Finally, apply VOTE function back to the first level of the ms-tree of all peers and a unanimity value is obtained.

Function *immigration-processing*:

- All peers exist in the network originally send the received value in the (r-1)th bout to the new peer.
- The new peer takes the majority values on the received values, and then stores these values to the level r-1 of its ms-tree.

Function *emigration-processing*:

- All peers rebuild the ms-tree by eliminating the values received from the left peers.

Fig. 1 The Algorithm BAMP2P.

3. Correctness of the algorithm

The following Lemmas, Corollary and Theorems are used to prove the correctness of the proposed protocol.

Lemma 1: Assume that α is a vertex, if the sub-tree rooted at α has a common frontier, α is common.

Proof: We need to prove the height of α by induction:

We can say that α is common, if α's height 0 and there exists a common frontier (α itself). Furthermore, when α's height is l, α's children will be the same according to the induction hypothesis result of the height as l-1; hence, we can say that vertex α is common.

Lemma 2: The values revised by the majority value of reliable peers are the same.

Proof: By Lemmas 1, 2, and 3, all the ms-trees' correct vertices are the same and all non-faulty peers' ms-trees also have the same common frontier. Besides, there is at least n-⌊(n-1)/3⌋ peers are non-faulty. Hence, all these non-faulty peers must be reliable peers. Thus, for the third level of the ms-tree, the majority value of these reliable peers' sub-tree must be the same. Hence, the values which are revised according to the majority value of the reliable peer will be in common.

Lemma 3: The new participant can use the majority value within the values received from others as the value received from the (r-1)th bout for exchanging message.

Proof: The new participant can user the majority value within the values which are received from other peers. Basically, there exists more than $n-\lfloor(n-1)/3\rfloor$ non-faulty peers and they will transmit the received values to the new participant honestly. This means that the network has more than $n-\lfloor(n-1)/3\rfloor$ correct values received by the new participant, and these correct values collectively signify that most of the non-faulty peers agree upon the majority value. Hence, using the majority value as the value which is received from previous bouts is correct.

Corollary 1: The root will be the same, if the ms-tree exists a common frontier.

Theorem 1: The roots in the ms-trees of the non-faulty peers are common.

Proof: By the results of Corollary 1, Lemma 2, and Lemma 3, the theorem is proven.

Theorem 2: BAMP2P can solve the BA problem.

Proof: For proving this theorem, we need to prove that BAMP2P can meet the following requirements:

- Agreement': The value s of the root is the same.

By the result of Theorem 1, this requirement is satisfied.

- Validity': If the starter is non-faulty, the VOTE(s) value will be equal to v_s for all non-faulty peers.

When the starter is non-faulty, it will broadcast the same value v_s to all other peers in the network. For all non-faulty peers' ms-tree, the correct vertices' value is v_s. Hence, all ms-trees' correct vertex are the same, and the value will be v_s. Since the starter is non-faulty, the root of the ms-tree is also a correct vertex. By Theorem 1, the root will be the same. The computed value VOTE(s) = v_s will be stored in the root for all non-faulty peers. Hence, Validity' is satisfied.

4. Conclusion

To improve the fault-tolerance problem under mobile P2P network, a novel agreement algorithm, called BAMP2P, has been proposed. The BAMP2P algorithm can use only three bouts for exchanging message to ensure that each non-faulty peer reaches an agreement no matter how many peers exist under the mobile P2P network. Our algorithm can decrease the complexity of message to $O(n^2)$. BAMP2P also thinks about the peers roaming issue for the protocol. Thus, BAMP2P can allow all non-faulty peers to obtain a common value and can tolerate $n-\lfloor(n-1)/3\rfloor$ faulty peers even if some peers roam among different mobile P2P network area. Therefore, BAMP2P is more suitable than previous works to ensure that all non-faulty peers can agree on an unanimity value under a mobile P2P network.

References

[1] http://www.bittorrent.com/

[2] C.F. Cheng and K. T. Tsai, Eventual Strong Consensus with Fault Detection in the Presence of Dual Failure Mode on Processors under Dynamic Networks, Journal of Network and Computer Applications, vol. 35, no. 4, pp.1260-1276, July 2012, Elsevier Press.

[3] K.Q. Ya, and S. C. Wan, Grouping Byzantine agreement, Computer Standard & Interfaces vol. 28, no. 1, pp. 75-92, 2005.

[4] K. Q. Yan, S. C. Wang, and M. L. Chiang, Optimal Agreement in a Scale-Free Network Environment, Informatica International Journal vol. 17, no. 1, pp. 137-150, 2006.

[5] L. Lamport, R. Shostak, and M. Pease, The Byzantine Generals Problem, ACM Transactions on Programming Languages and Systems, vol. 4, no. 3, pp. 382-401, 1982.

[6] M. Pease, R. Shostak, and L. Lamport, Reaching Agreement in Presence of Faults, Journal of ACM vol. 27, no. 2 pp. 228-234, 1980.

[7] M. Correia, A. N. Bessani, and P. Veríssimo, On Byzantine generals with alternative plans, Journal of Parallel and Distributed Computing vol. 68, no. 9, pp. 1291-1296, 2008.
[8] M. Fischer, and N. Lynch, A Lower Bound for the Assure Interactive Consistency, Information Processing Letters vol. 14, no. 4, pp. 183-186, 1982.
[9] N. Natalya, and L. Veltri, Byzantine Generals Problem in the Light of P2P Computing, 3rd Annual International Conference on Mobile and Ubiquitous Systems (Workshops), pp. 1-5, 2006.
[10] S. C. Wang, K.Q. Yan, S. S. Wang, and G. Y. Zheng, Reaching Agreement among Virtual Subnets in Hybrid Failure Mode, IEEE Trans. Parallel and Distributed Systems vol. 19, no. 9, pp. 1252-1262, 2008.
[11] W. Zhao, Design and implementation of a Byzantine fault tolerance framework for Web services, Journal of Systems and Software vol. 82, no. 6, pp. 1004-1015, 2009